Universidad de Salamanca

#### IHMI IB Y OBJECTIVE-C

Depto. Informática y Automática Máster en Sistemas Inteligentes Dr. J.R. García-Bermejo Giner

#### Tercera parte - Introducción

La interfaz gráfica de usuario que ofrece Mac OS X™ es una referencia para la industria.

Está basada en dos pilares fundamentales: Objective-C (q. v.) y Cocoa, una colección de clases escritas en C/Objective-C.

El entorno integrado de desarrollo XCode está complementado con un sofisticado entorno dedicado a la generación de interfaces gráficas de usuario (Interface Builder). Estas interfaces gráficas de usuario se serializan y almacenan en disco, siendo después utilizadas por las aplicaciones ya terminadas.

La arquitectura de las aplicaciones escritas en Cocoa es la conocida Modelo-Vista-Controlador. La página web del creador de esta arquitectura se encuentra aquí: Tryve Reenskauq

Relación entre Interface Builder y Objective-C

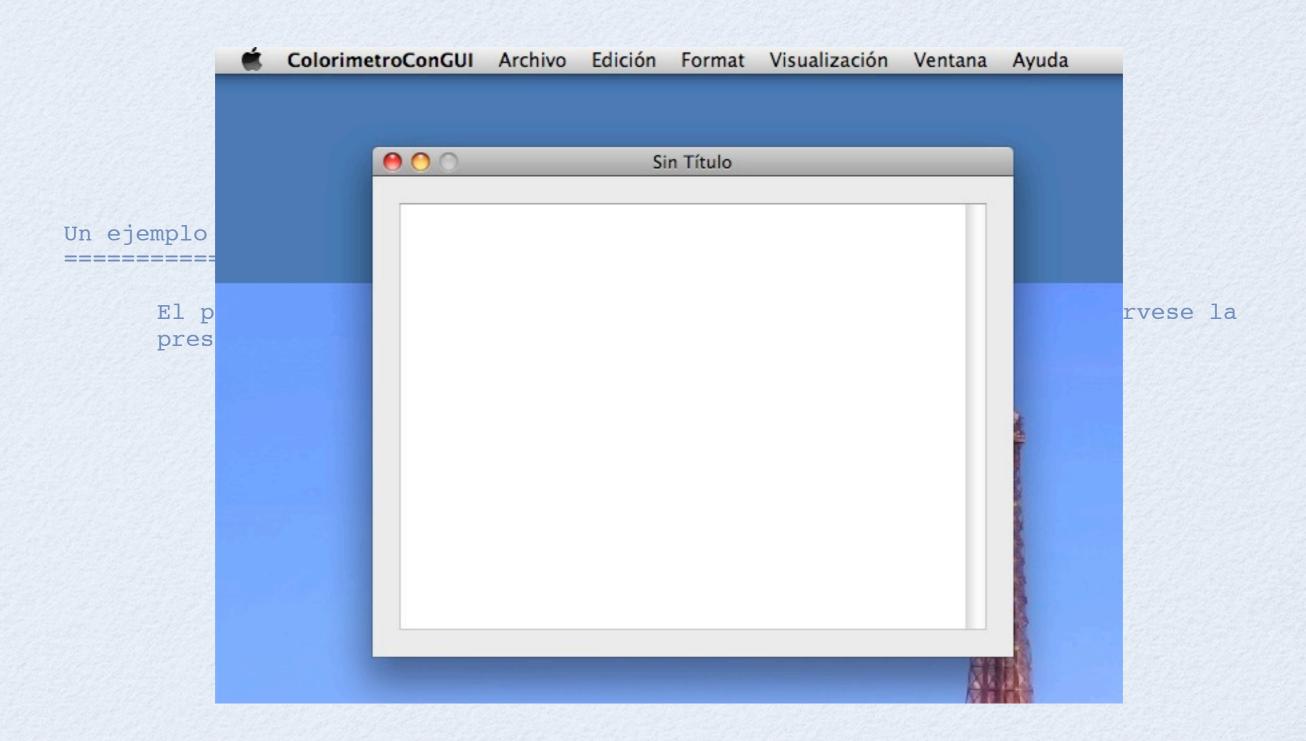
Vamos a estudiar brevemente el mecanismo de creación de aplicaciones en Cocoa

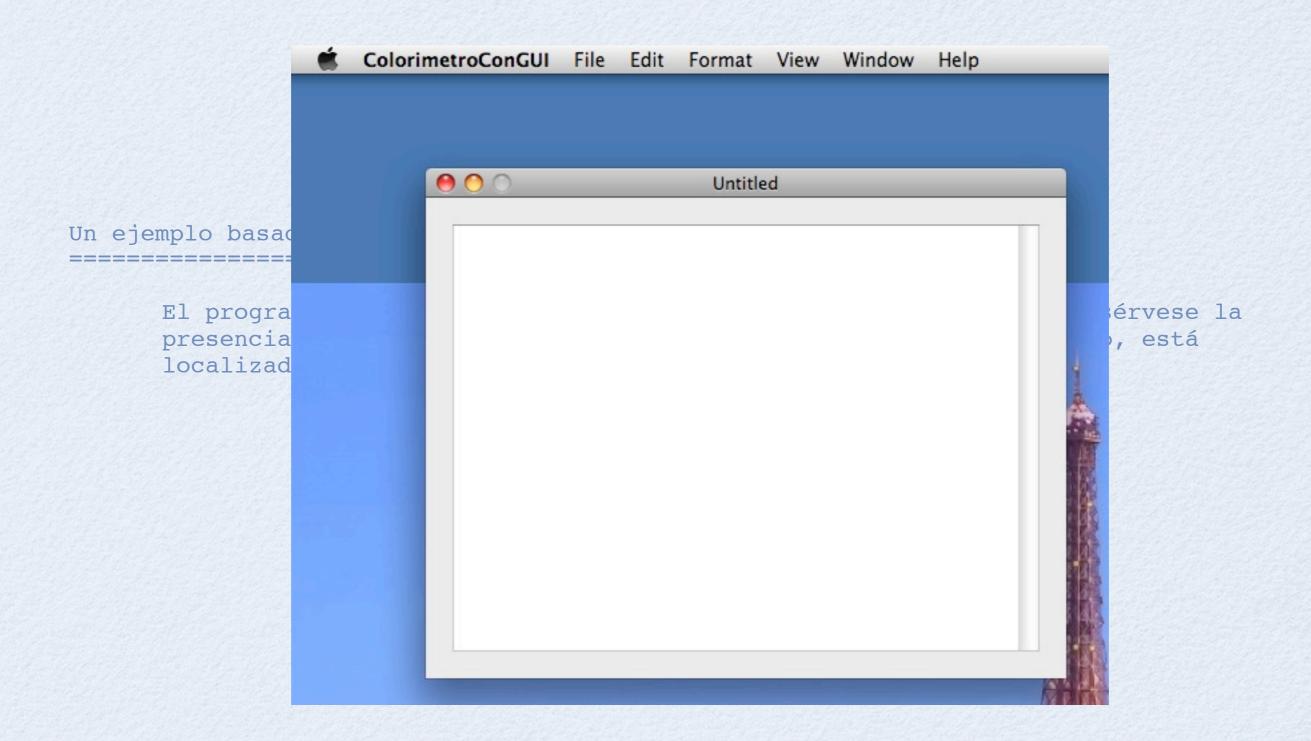
- Uso de Interface Builder.
- Uso de XCode (con Objective-C) en conjunción con Interface Builder.

Un ejemplo basado en Cocoa

La clase construida en los ejercicios anteriores, ListaDeAlimentos, sirve como punto de partida para un programa que hace de forma gráfica lo mismo que se hacía anteriormente en forma de texto.

Ha sido preciso realizar algunas modificaciones, aunque el grueso de las novedades se concentra en la clase denominada Controlador. Los métodos empleados en la clase ListaDeAlimentos sufren cambios, como era de esperar. Un programa escrito en C/C++ se integra de forma natural con una interfaz gráfica de usuario escrita en Cocoa.





```
BasicXML.h
========

#import <Cocoa/Cocoa.h>

@protocol BasicXML

@required
-(B00L)readFromXML:(NSString *)theXMLfile;
-(B00L)writeToXML:(NSString *)theXMLfile;
@optional
-(void)allKeysUppercase:(B00L)yesorno;

@end
```

```
ListaDeAlimentos.h
#import <Cocoa/Cocoa.h>
#import <AppKit/NSGraphics.h>
#import "BasicXML.h"
@interface ListaDeAlimentos : NSObject <BasicXML> {
@private
    NSMutableDictionary * lista;
    NSString * nombreDelArchivo;
}
@property(readonly) NSMutableDictionary * lista;
@property(readwrite, copy) NSString * nombreDelArchivo;
-(id)init;
-(float)caloriasPorCienGramosDe:(NSString*)alimento;
-(void)registrar:(NSString*)alimento caloriasPorCienGramos:(float)valor;
-(void)eliminar:(NSString*)alimento;
@end
```

```
ListaDeAlimentos.m
______
#import "ListaDeAlimentos.h"
@implementation ListaDeAlimentos
@synthesize lista;
@synthesize nombreDelArchivo;
-(id)init { ... }
-(float)caloriasPorCienGramosDe:(NSString*)alimento { ... }
-(void)registrar:(NSString*)alimento caloriasPorCienGramos:(float)valor { ... }
-(void)eliminar:(NSString*)alimento { ... }
-(BOOL)readFromXML:(NSString *)theXMLfile{ ... }
-(BOOL)writeToXML:(NSString *)theXMLfile{ ... }
@end
```

```
Controller.h
@interface Controller : NSObject {
    @private
    IBOutlet NSTextView * t;
    IBOutlet NSWindow * v;
    ListaDeAlimentos * lda;
    NSBundle * main;
}
@property(readonly) ListaDeAlimentos *lda;
                                              @property(readonly) NSTextView * t;
@property(readonly) NSWindow * v;
                                               @property(readonly) NSBundle * main;
-(id)init;
-(IBAction)newFile:(id)sender;
-(IBAction)open:(id)sender;
-(IBAction)close:(id)sender;
-(IBAction)save:(id)sender;
-(IBAction)saveAs:(id)sender;
-(IBAction)revertToSaved:(id)sender;
-(IBAction)importFromXML:(id)sender;
-(IBAction)exportToXML:(id)sender;
@end
```

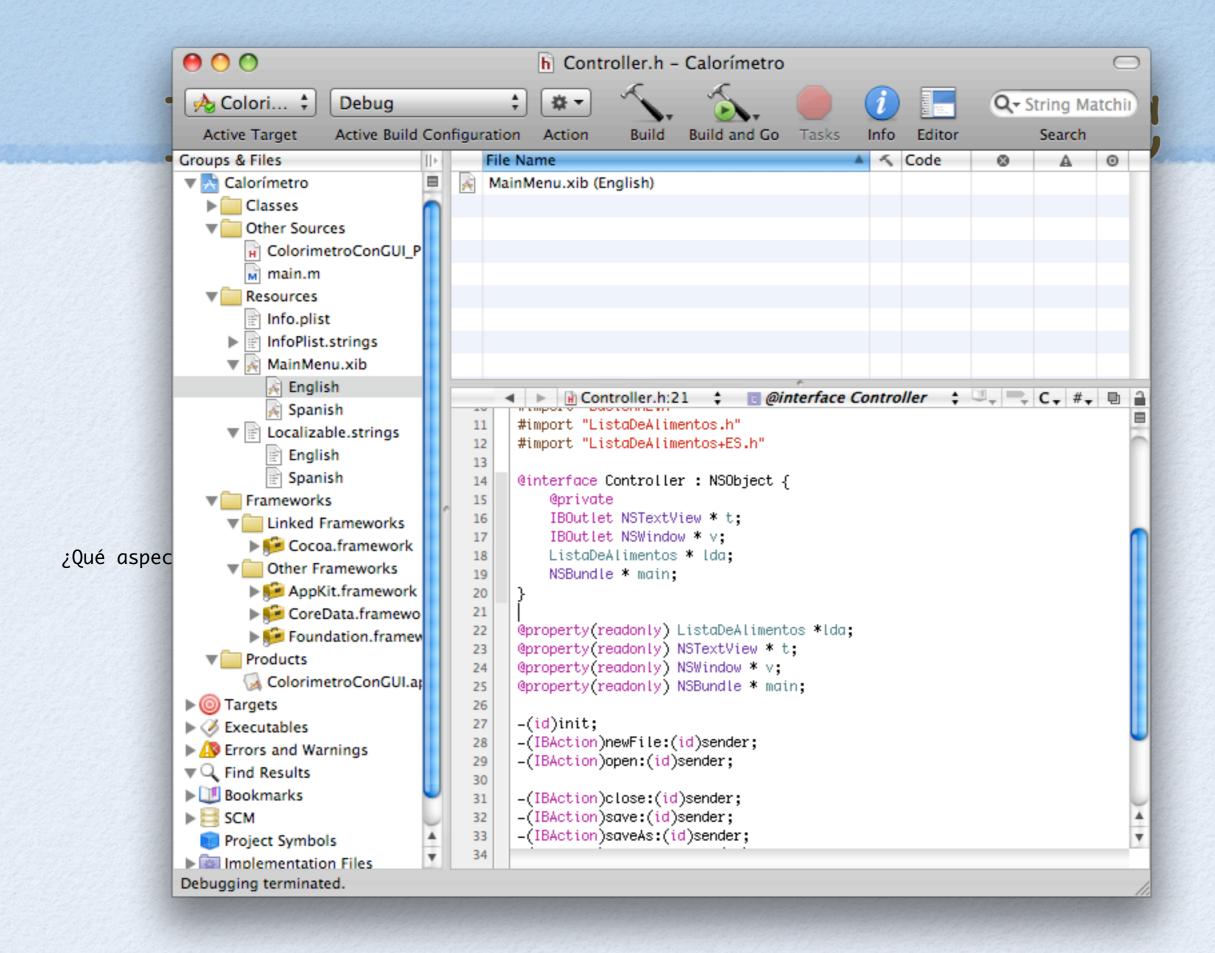
```
Controller.m
_____
#import "Controller.h"
@implementation Controller
@synthesize lda;
@synthesize t;
@synthesize v;
@synthesize main;
-(id)init{ }
-(id)awakeFromNib { }
-(IBAction)newFile:(id)sender{ }
-(IBAction)open:(id)sender{ }
-(IBAction)close:(id)sender{}
-(IBAction)save:(id)sender{ }
-(IBAction)saveAs:(id)sender{}
-(IBAction)revertToSaved:(id)sender{ }
-(IBAction)importFromXML:(id)sender{ }
-(IBAction)exportToXML:(id)sender{ }
-(BOOL)validateMenuItem:(NSMenuItem*)menuItem{}
```

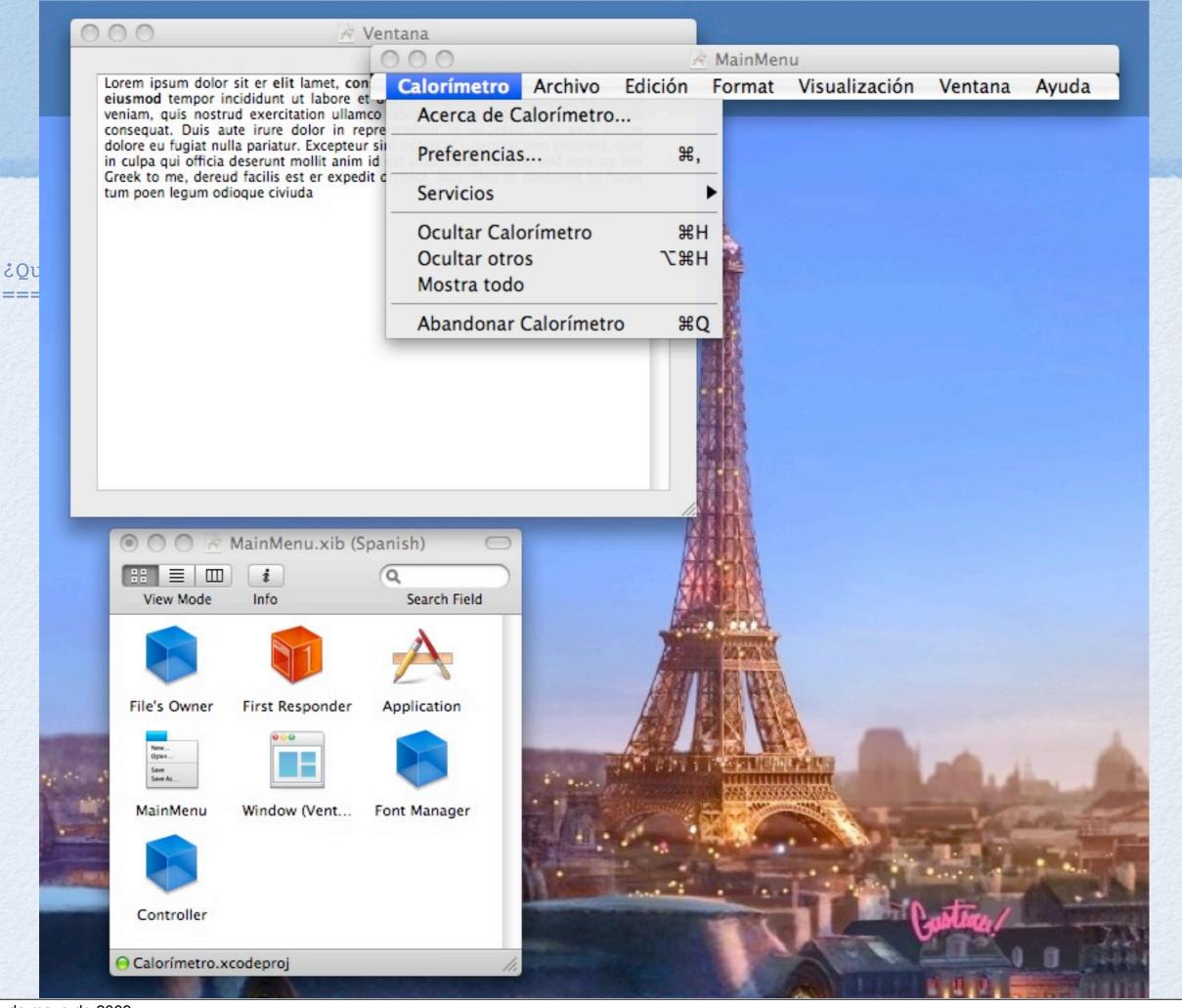
@end

```
main.m
======

#import <Cocoa/Cocoa.h>

int main(int argc, char *argv[])
{
    return NSApplicationMain(argc, (const char **) argv);
}
```





# 

#### Comentarios

Aun cuando requiere un cierto esfuerzo, es posible desarrollar (también en equipo) aplicaciones con sofisticadas interfaces de usuario, aplicando la tecnología orientada a objetos y la arquitectura MODELO-VISTA-CONTROLADOR.

Se aprecia una tendencia a interfaces de tipo táctil y visual, en que los gestos realizados manualmente aportan un nivel de interacción muy superior a habido hasta el momento.

La portabilidad de dispositivos ofrece posibilidades muy interesantes al combinar un tamaño reducido con la posibilidad de acceder via red a la misma información que un ordenador.

Nota final: el entorno que se ha mostrado a lo largo del curso sirve para desarrollar aplicaciones para Mac OS X. También es el entorno de desarrollo de iPhone.

#### Conclusiones

La elaboración de una interfaz gráfica de usuario puede apoyarse en el mecanismo de serialización que aportan distintos lenguajes orientados a objetos.

La Interacción Hombre-Máquina pasa por una fase de gran interés, en que los mecanismos propios de la Web tienen una relación directa con los mecanismos propios del escritorio.

El uso de animaciones es el efecto del momento, y lo será a medio plazo.

Los dispositivos móviles empiezan a tener características propias de ordenadores portátiles o de sobremesa.

Los dispositivos de sobremesa aportan sus características a los dispositivos móviles.

En un plazo de 5 años, no se concebirá un dispositivo móvil sin E/S de vídeo, y capaz de acceder a la misma información que hoy requiere un ordenador.

JUST DO IT!

#### IHM - Java

El paso siguiente --->

Apenas hemos entreabierto la caja de Pandora. El lector interesado puede encontrar aquí caminos que le lleven en cualquier dirección, teniendo como Norte facilitar la comunicación entre Hombre y Máquina. Ya se ha avanzado mucho, pero solo podemos estar seguros de una cosa: por sorprendente que pueda parecer el futuro, la realidad será siempre mucho más extraordinaria.

Salamanca, Junio de 2008 J.R. García-Bermejo Giner

Universidad de Salamanca

# IHMINA IB Y OBJECTIVE-C

Depto. Informática y Automática Máster en Sistemas Inteligentes Dr. J.R. García-Bermejo Giner